
AWS
re:Invent

ARC316

Testing resiliency using chaos engineering

Seth Eliot

Principal Reliability Solutions Architect, AWS Well-Architected

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“We needed to build systems that embrace failure as a natural occurrence.”



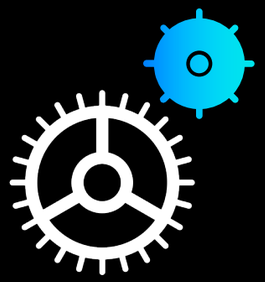
Werner Vogels

CTO

Amazon.com

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Well-Architected Framework



Operational excellence



Security



Reliability



Performance efficiency



Cost optimization

Testing resiliency

Resiliency

"The ability of a system to recover from infrastructure or service disruptions…"

Design principles for reliability

- Automatically recover from failure
- Test recovery procedures

http://bit.ly/reliability-pillar



Automatically recover from failure



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Fault isolation to protect your workload

AWS Well-Architected Framework

Add a link to your architectural design

REL 10. How do you use fault isolation to protect your workload? Info

Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.

Question does not apply to this workload Info

Select from the following

- [x] Deploy the workload to multiple locations Info
- [ ] Automate recovery for components constrained to a single location Info



https://bit.ly/use-fault-isolation

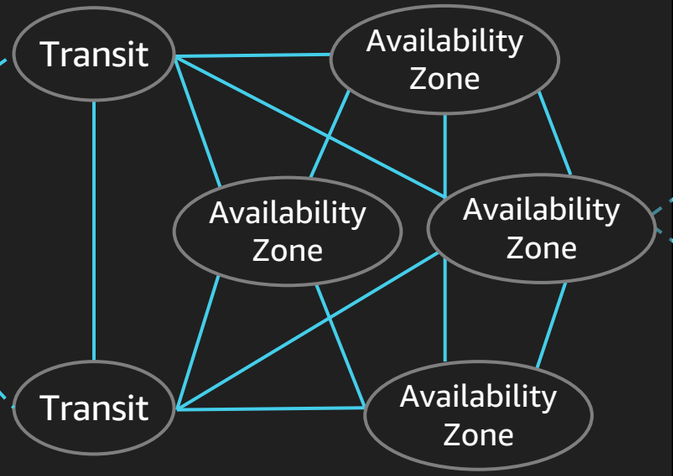
AWS Regions and Availability Zones

AWS REGIONS ARE PHYSICAL LOCATIONS AROUND THE WORLD WHERE WE CLUSTER DATA CENTERS

24 AWS Regions worldwide

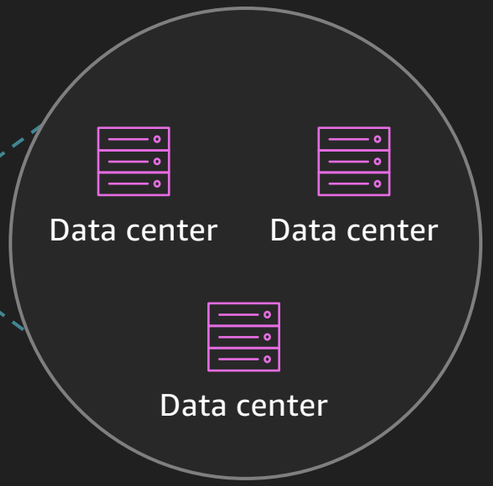


Each AWS Region has multiple Availability Zones



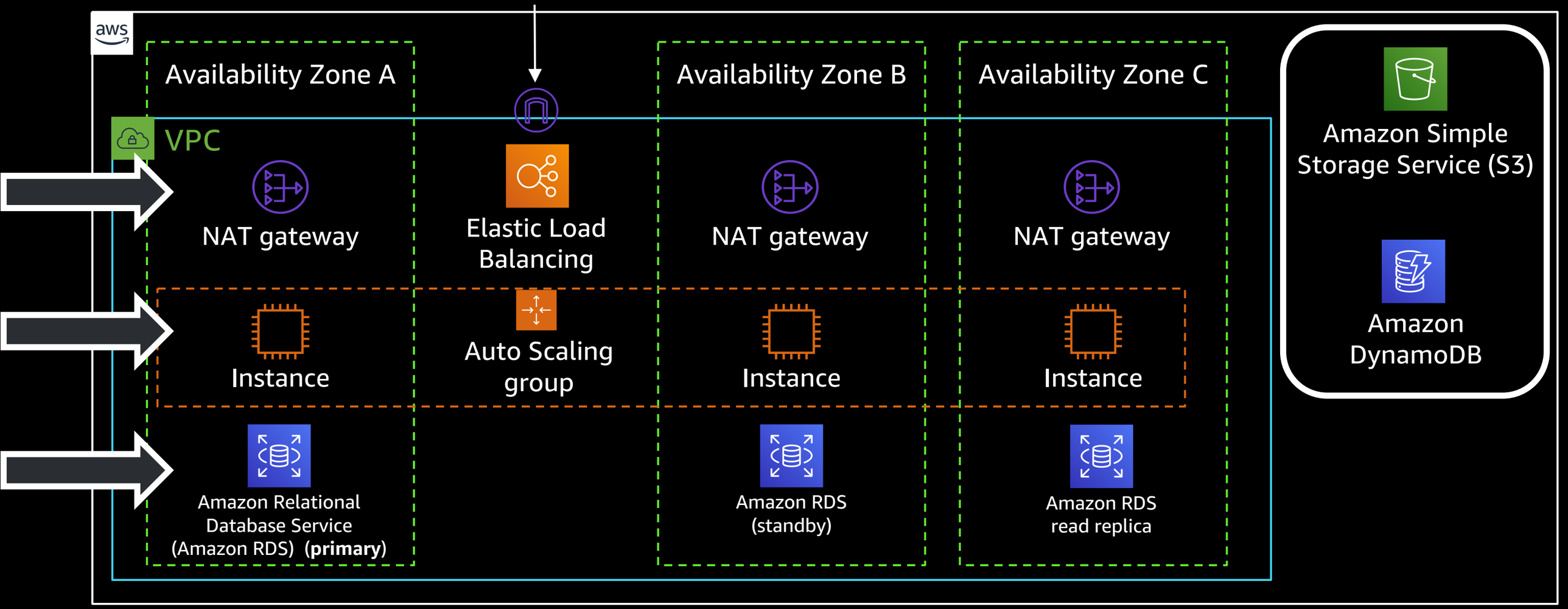
A **Region** is a physical location in the world

Each Availability Zone is one or more discrete data centers

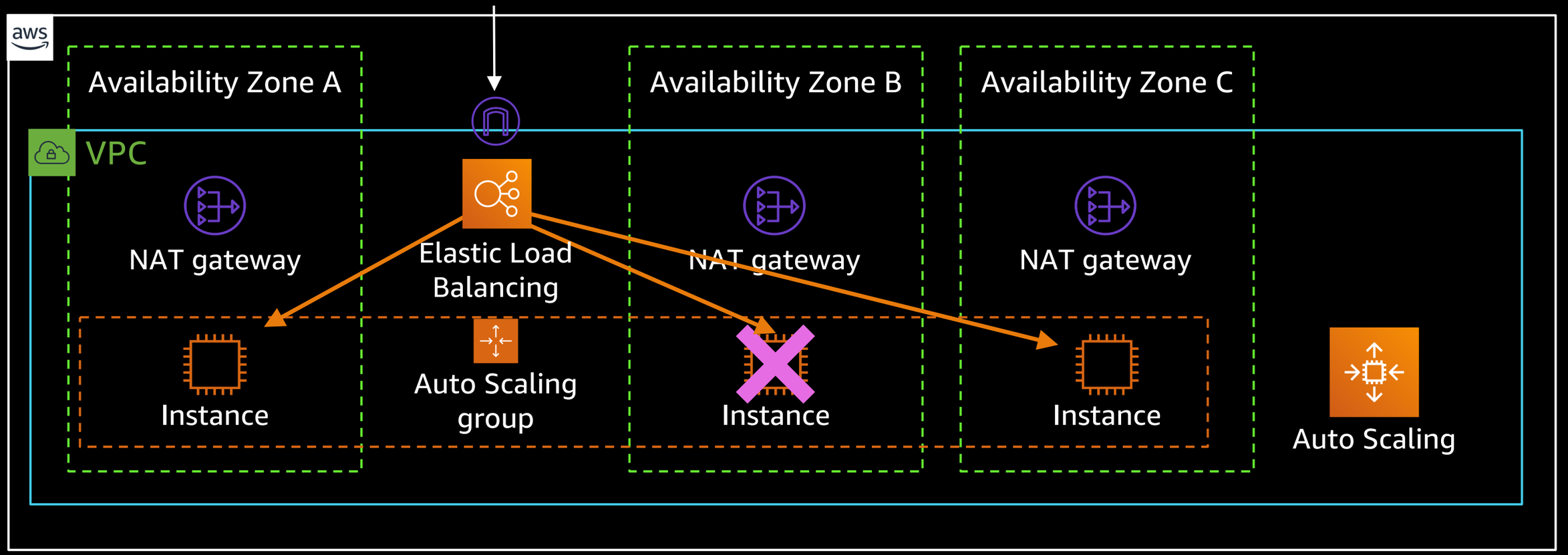


Data centers, each with redundant power, networking, and connectivity, housed in separate facilities

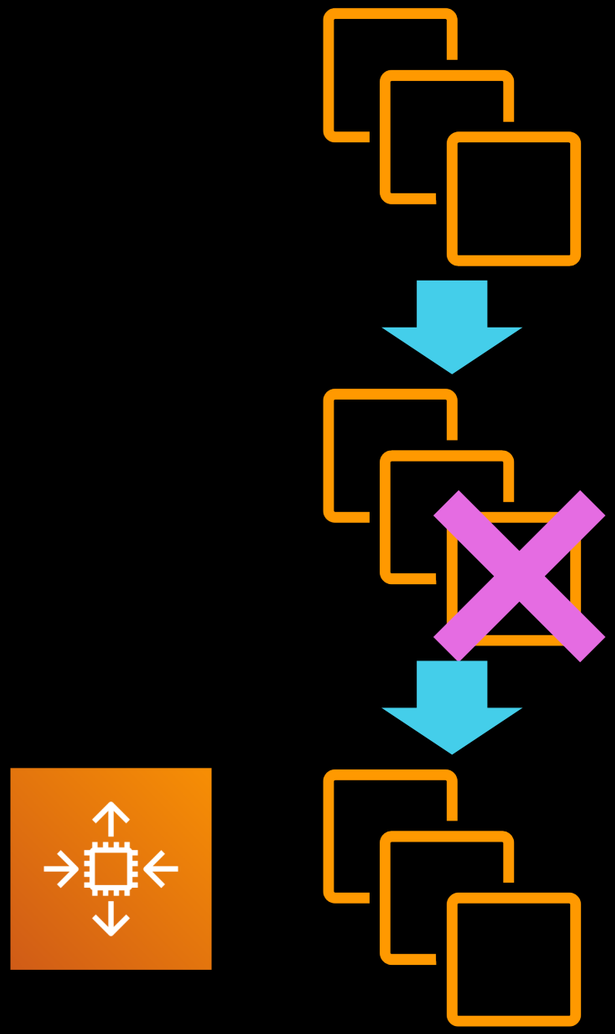
Three tiers, Multi-AZ



Elastic Load Balancing and auto-recovery



AWS Auto Scaling and auto-recovery



REL 5. How do you design interactions in a distributed system to mitigate or withstand failures? Info

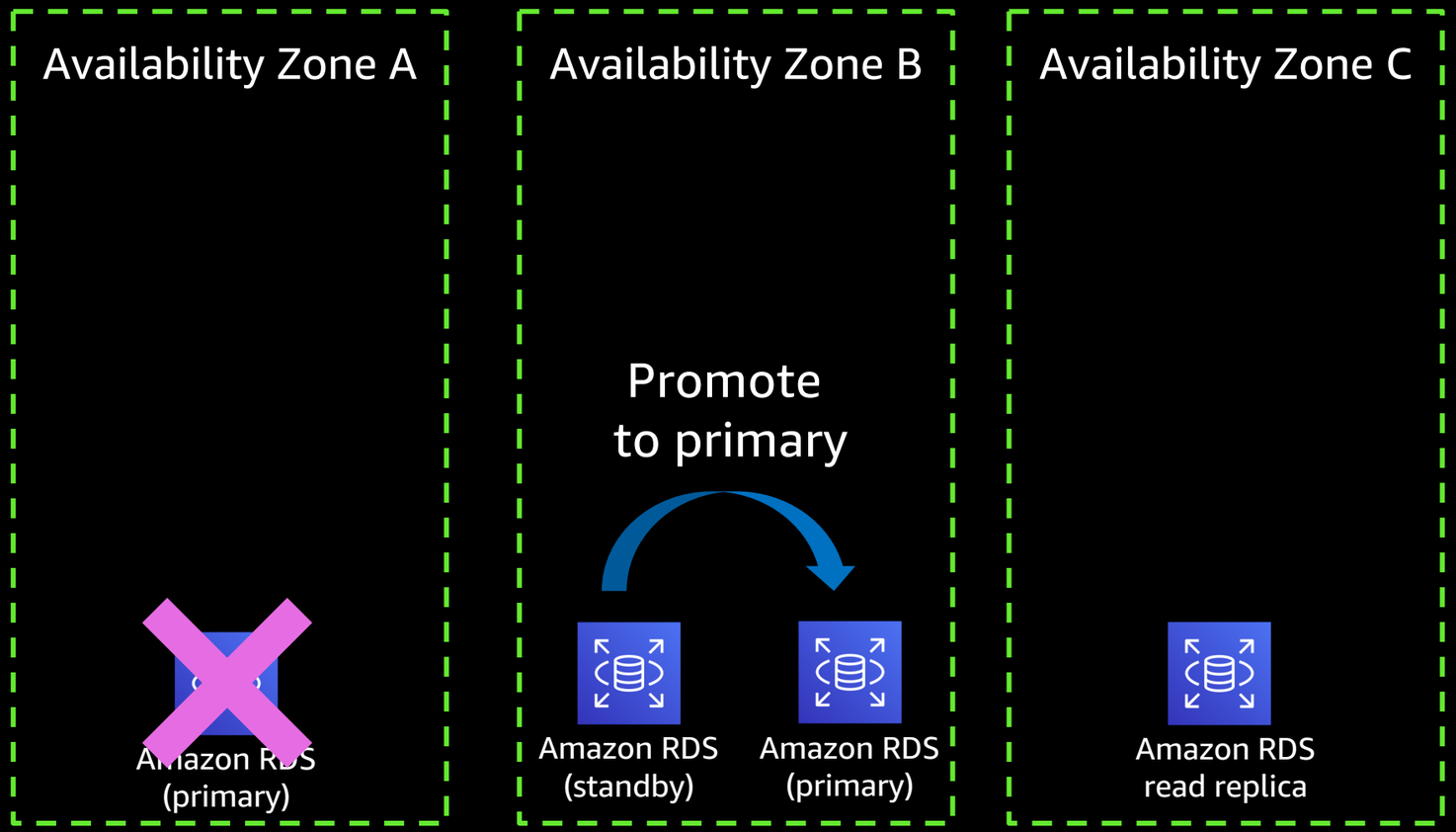
Distributed systems rely on communications networks to interconnect components (such as servers or services). Your workload must operate reliably despite data loss or latency over these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices enable workloads to withstand stresses or failures, more quickly recover from them, and mitigate the impact of such impairments. The result is improved mean time to recovery (MTTR).

Question does not apply to this workload Info

Select from the following

- [x] Implement graceful degradation to transform applicable hard dependencies into soft dependencies Info
- [x] Throttle requests Info
- [x] Control and limit retry calls Info
- [x] Fail fast and limit queues Info
- [x] Set client timeouts Info
- [x] Make services stateless where possible Info

Amazon RDS and auto-recovery



Amazon RDS

Amazon Aurora

Does not have standby

Can promote *any*
read replica to primary

Testing resiliency with chaos engineering



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production.”



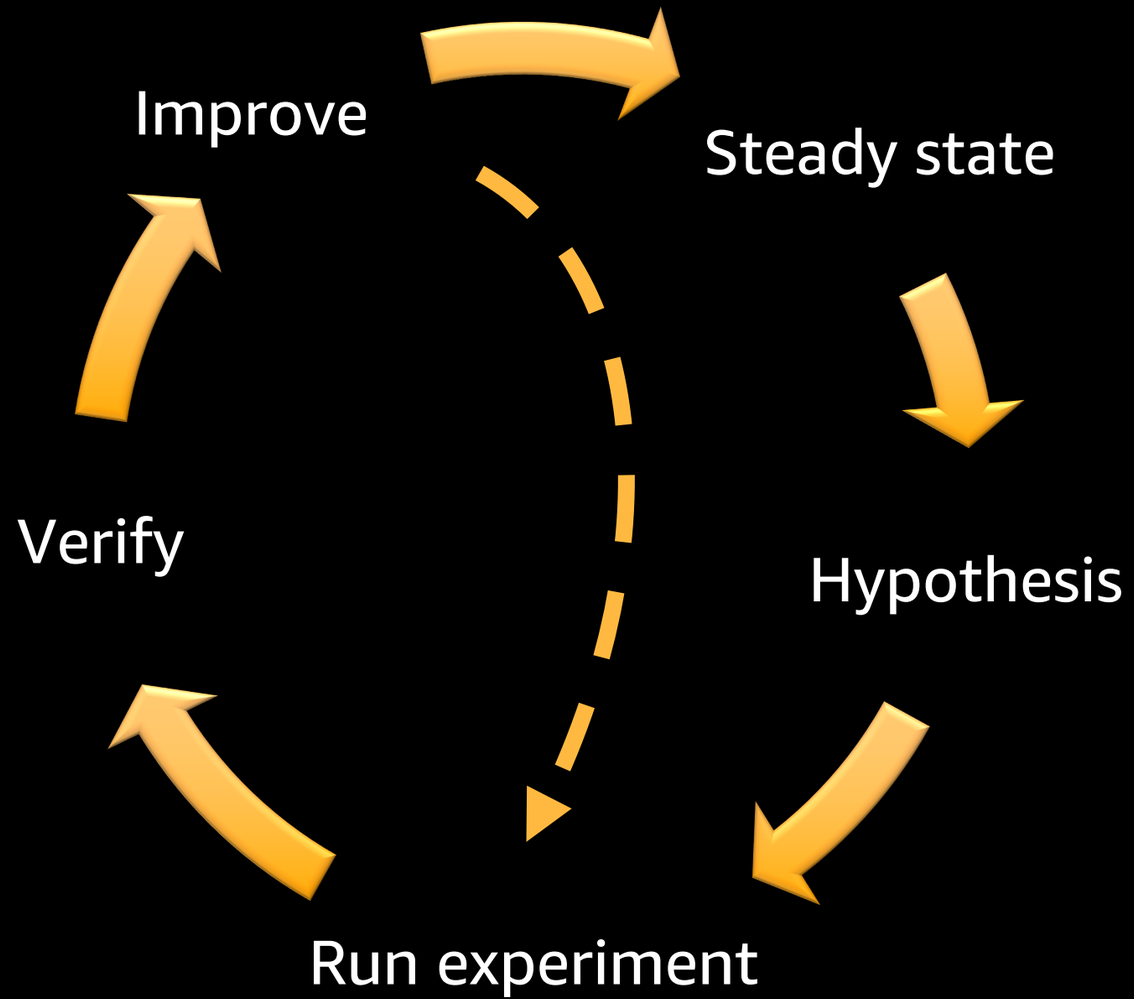
“Principles of Chaos Engineering”

http://principlesofchaos.org

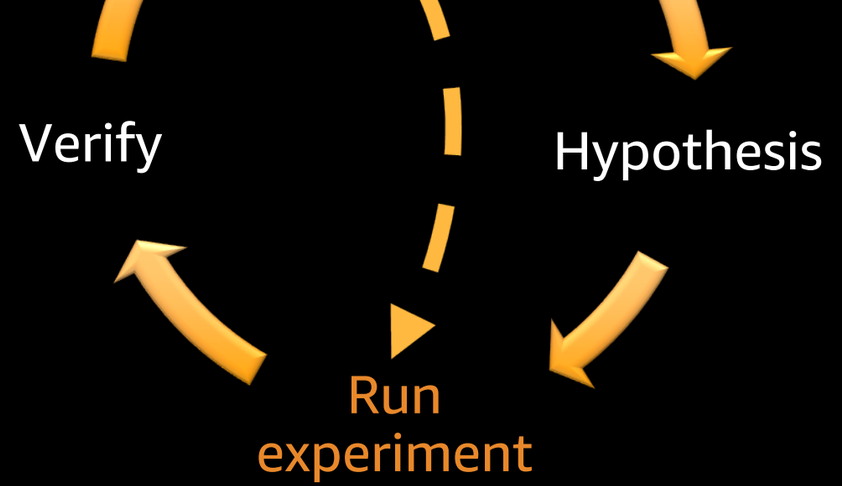
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Chaos engineering

A SCIENTIFIC METHOD



Chaos experiment



Inject events that simulate

- Hardware failures, like servers dying
- Software failures, like malformed responses
- Nonfailure events, like spikes in traffic or scaling events

Any event capable of disrupting steady state





Chaos engineering is not new

2005

Amazon Game Days
Train how to respond to disaster



2010

Netflix Chaos Monkey



2016

Principles of Chaos
Chaos Engineering



Tools for the job

Plenty of open-source and commercial tools

Just use these?

Yes! Absolutely. Please do.



Let’s take a walk here and now, and then you can run

- You can write code to simulate these events in any language
- The lab includes Bash shell, Python, Java, PowerShell, and C#

- You can write your own Availability Zone failure simulation

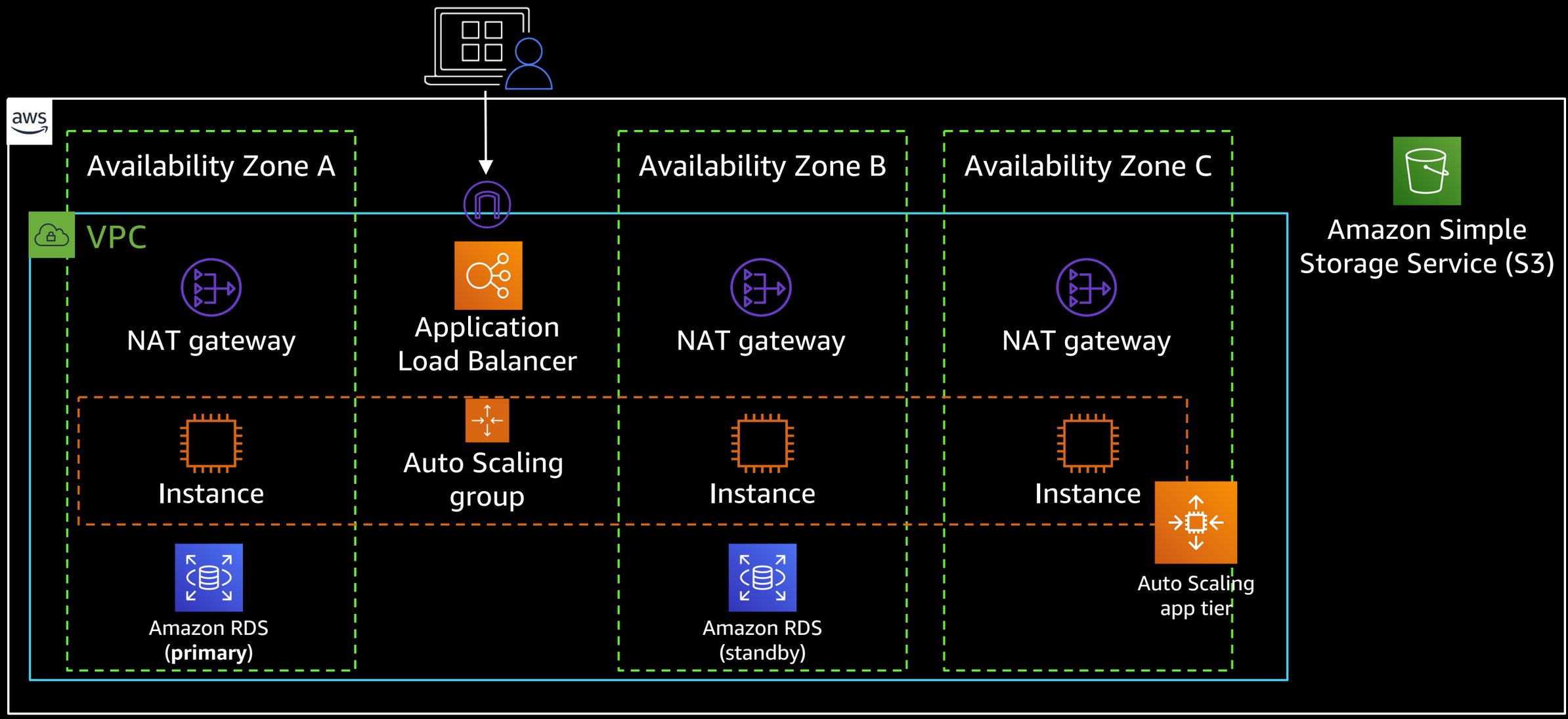
Experiment: Component and Availability Zone failure



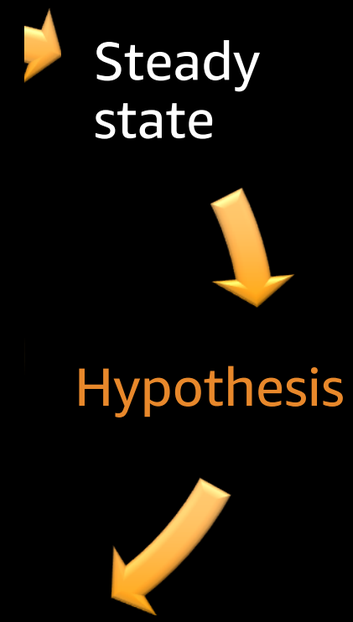
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our architecture for the demo



Hypotheses



Hypothesis 1
If one EC2 instance dies, then availability will not be impacted

Hypothesis 2
If an entire Availability Zone dies, then availability will not be impacted

- Load will be adequately served by remaining two EC2 instances
- Elastic Load Balancing will detect unhealthy instance and reroute requests
- Autoscaling will replace unhealthy instance
  - For Hypothesis 1, in same Availability Zone
  - For Hypothesis 2, in one of the other two Availability Zones

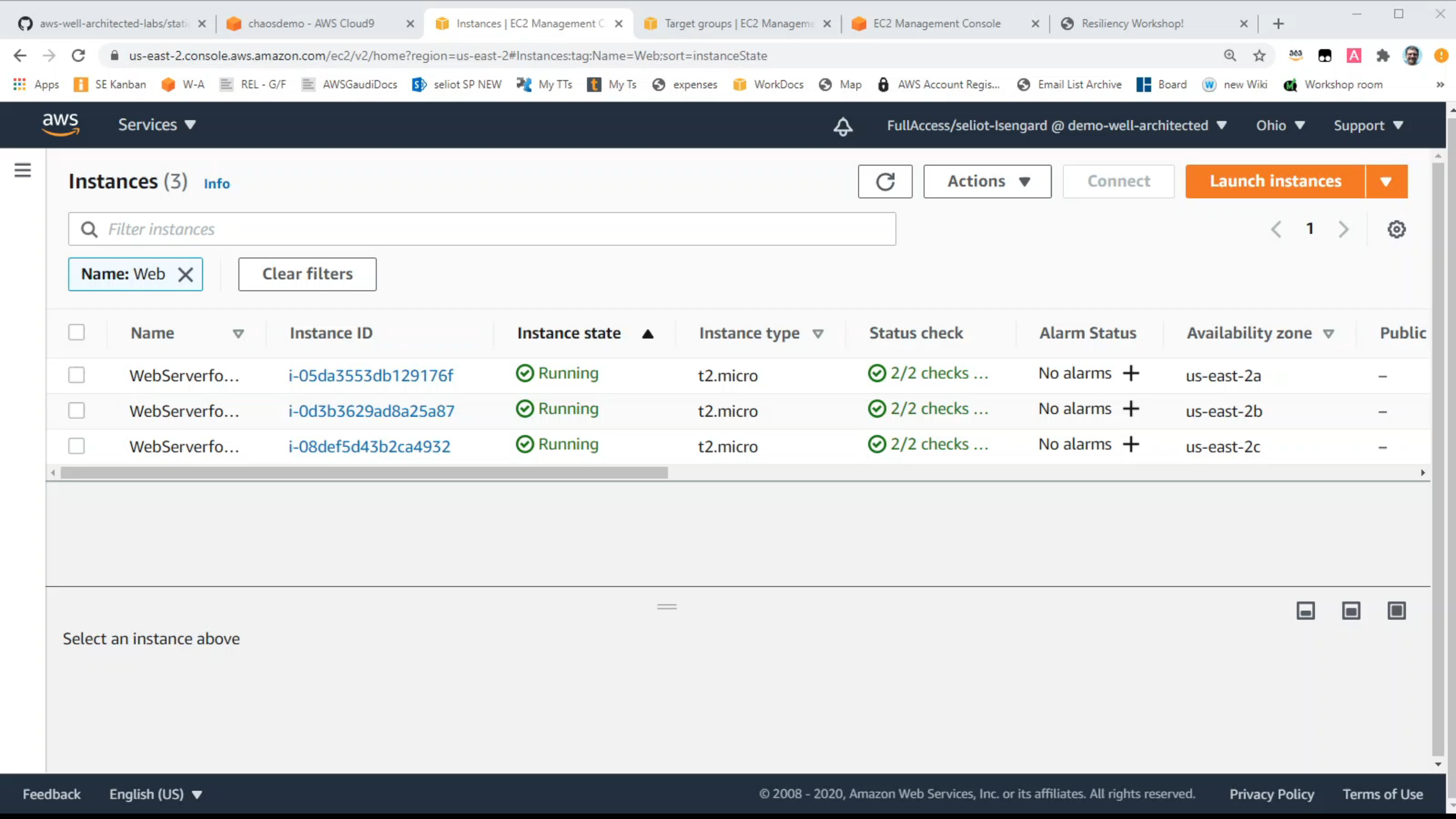
Demo: Component and Availability Zone failure experiment

http://bit.ly/wa-chaos



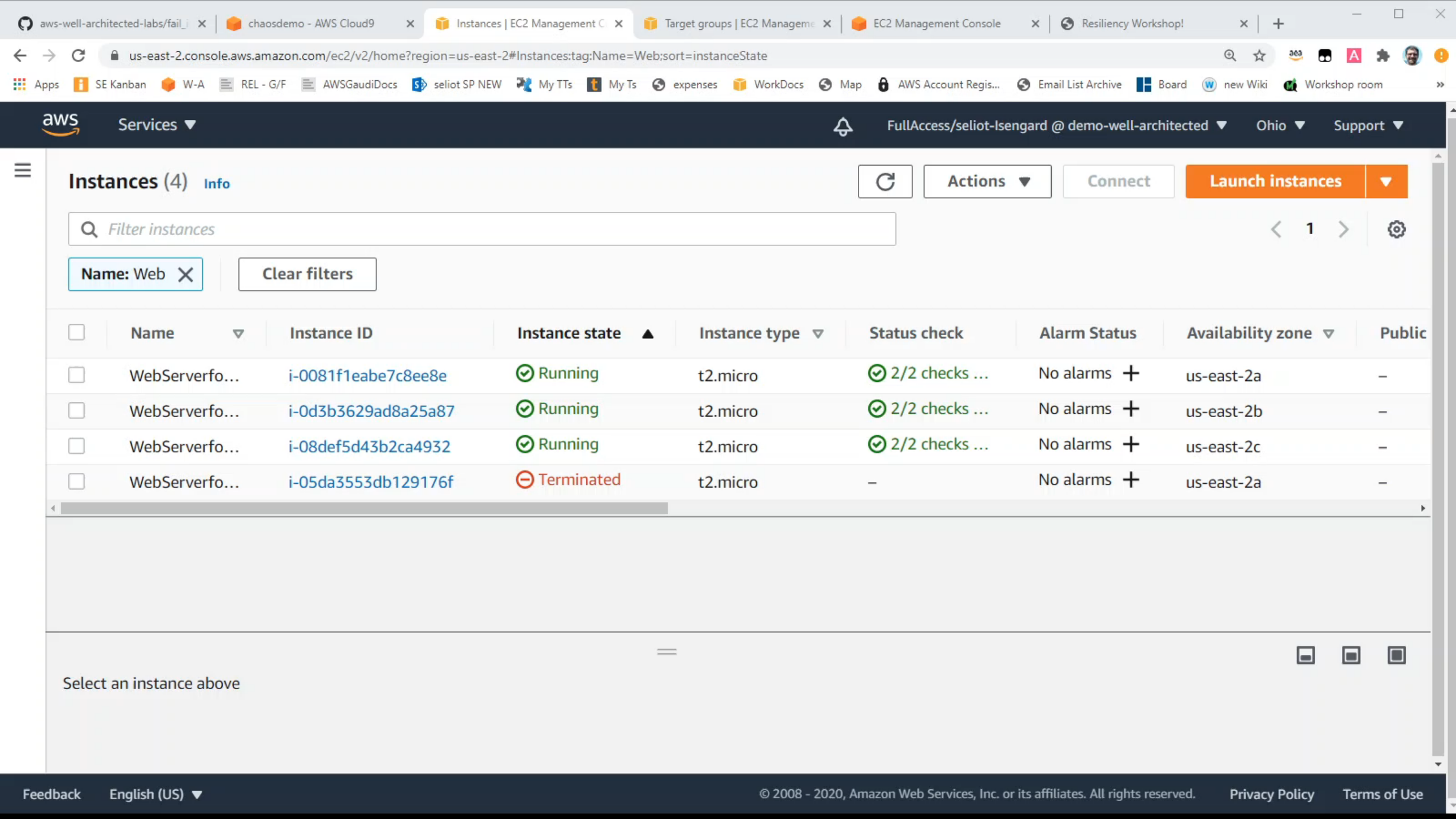
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


aws-well-architected-labs/stati
chaosdemo - AWS Cloud9
Instances | EC2 Management C
Target groups | EC2 Manageme
EC2 Management Console
Resiliency Workshop!
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Instances:tag:Name=Web;sort=instanceState
Apps
SE Kanban
W-A
REL - G/F
AWSGaudiDocs
seliot SP NEW
My TTs
My Ts
expenses
WorkDocs
Map
AWS Account Regis...
Email List Archive
Board
new Wiki
Workshop room
aws
Services
FullAccess/seliot-Isengard @ demo-well-architected
Ohio
Support
Instances (3) Info
Actions
Connect
Launch instances
Filter instances
1
Name: Web
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm Status
Availability zone
Public
WebServerfo...
i-05da3553db129176f
Running
t2.micro
2/2 checks ...
No alarms
us-east-2a
WebServerfo...
i-0d3b3629ad8a25a87
Running
t2.micro
2/2 checks ...
No alarms
us-east-2b
WebServerfo...
i-08def5d43b2ca4932
Running
t2.micro
2/2 checks ...
No alarms
us-east-2c
Select an instance above
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

One minute later...




aws-well-architected-labs/fail_
chaosdemo - AWS Cloud9
Instances | EC2 Management C
Target groups | EC2 Manageme
EC2 Management Console
Resiliency Workshop!
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Instances:tag:Name=Web;sort=instanceState
Apps
SE Kanban
W-A
REL - G/F
AWSGaudiDocs
seliot SP NEW
My TTs
My Ts
expenses
WorkDocs
Map
AWS Account Regis...
Email List Archive
Board
new Wiki
Workshop room
aws
Services
FullAccess/seliot-Isengard @ demo-well-architected
Ohio
Support
Instances (4) Info
Actions
Connect
Launch instances
Filter instances
1
Name: Web
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm Status
Availability zone
Public
WebServerfo... i-0081f1eabe7c8ee8e Running t2.micro 2/2 checks ... No alarms us-east-2a –
WebServerfo... i-0d3b3629ad8a25a87 Running t2.micro 2/2 checks ... No alarms us-east-2b –
WebServerfo... i-08def5d43b2ca4932 Running t2.micro 2/2 checks ... No alarms us-east-2c –
WebServerfo... i-05da3553db129176f Terminated t2.micro – No alarms us-east-2a –
Select an instance above
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Failure event code

fail_instance.sh

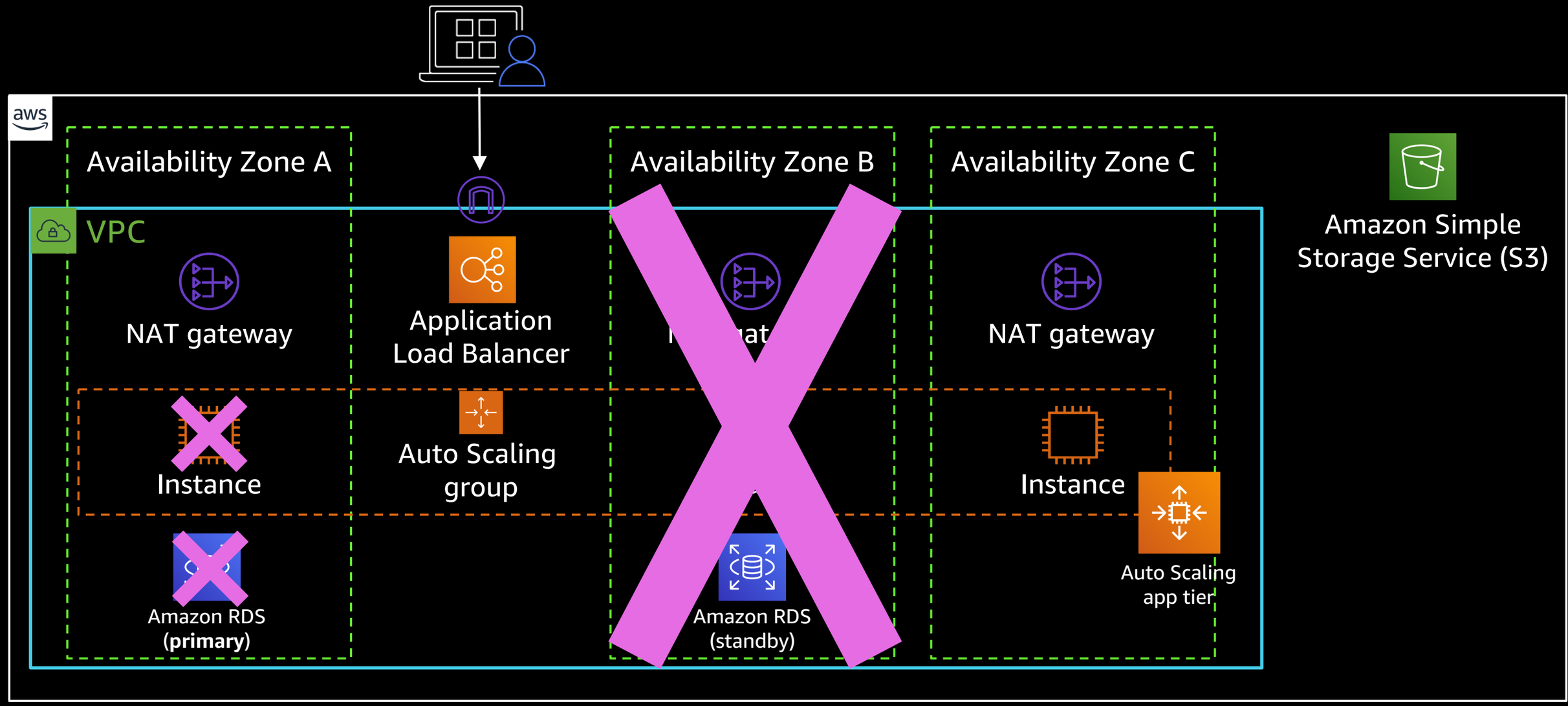
```
# Terminate that instance
aws ec2 terminate-instances --instance-ids $instance_id
```

fail_az.sh

```
# Add a network ACL on the subnets in the AZ to block all traffic
aws ec2 create-network-acl-entry <...> --cidr-block "0.0.0.0/0"
--egress  --protocol all --port-range From=0,To=65535 --rule-action deny

aws ec2 create-network-acl-entry <...>
--ingress <...>
```

Simulated failure experiments



You can run this as a lab

All the instructions are here



http://bit.ly/wa-chaos

You can access source code from this link on the lab



How this lab simulates failure events

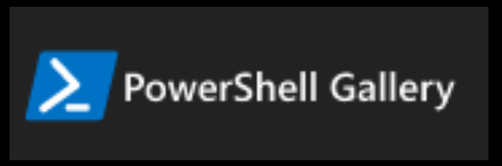
AWS Command Line Interface



AWS SDK



AWS Tools for PowerShell



Here is one more

AWS Systems Manager (SSM)



https://github.com/adhorn/chaos-ssm-documents

Making it real



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Reality…what a concept.”

Robin Williams



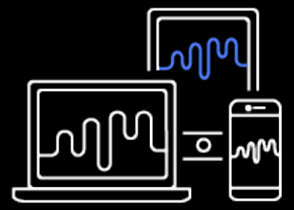
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reality is production with actual users

Your recovery design is based on *assumptions*

Reality may differ

Test close to production



Traffic patterns

Record / replay

Shadow testing

Test in production



Environment

Cloud-deployed

Test in production



Events

Are simulations

Learn from production

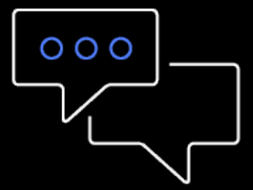
Game days

SIMULATE FAILURE OR EVENT TO TEST SYSTEMS, PROCESSES, AND TEAM RESPONSES



People

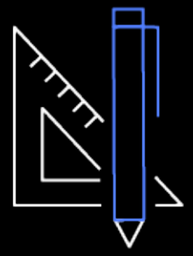
Cross-discipline team



Briefing

Overview

Roles

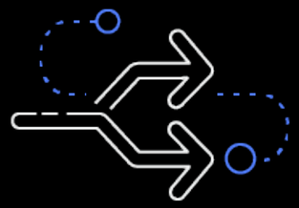


Planning

Scenario

Events

Preparation



Execution

In production

Capture feedback



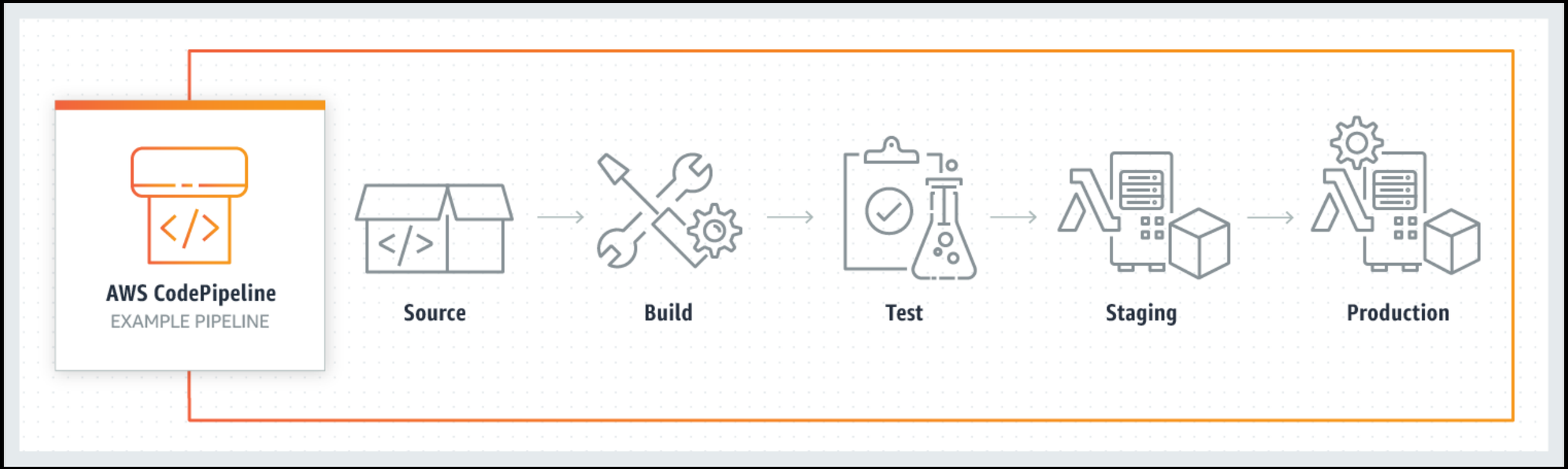
Analysis

What happened

Follow-up items

Run these tests regularly

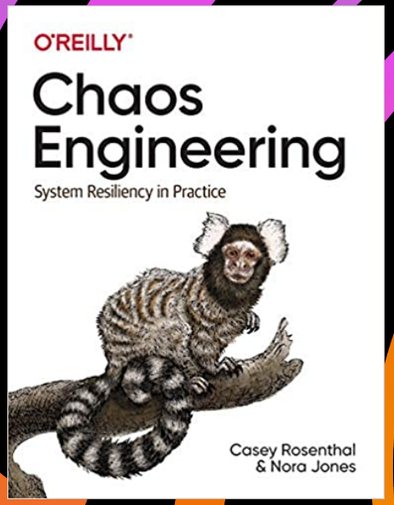
- As part of CI / CD pipeline
- Start with staging (test, pre-production) environments
- Consider which tests to run in production



“This isn’t about creating chaos. It is about making the chaos inherent in the system visible.”

Casey Rosenthal and Nora Jones

Chaos Engineering: System Resiliency in Practice

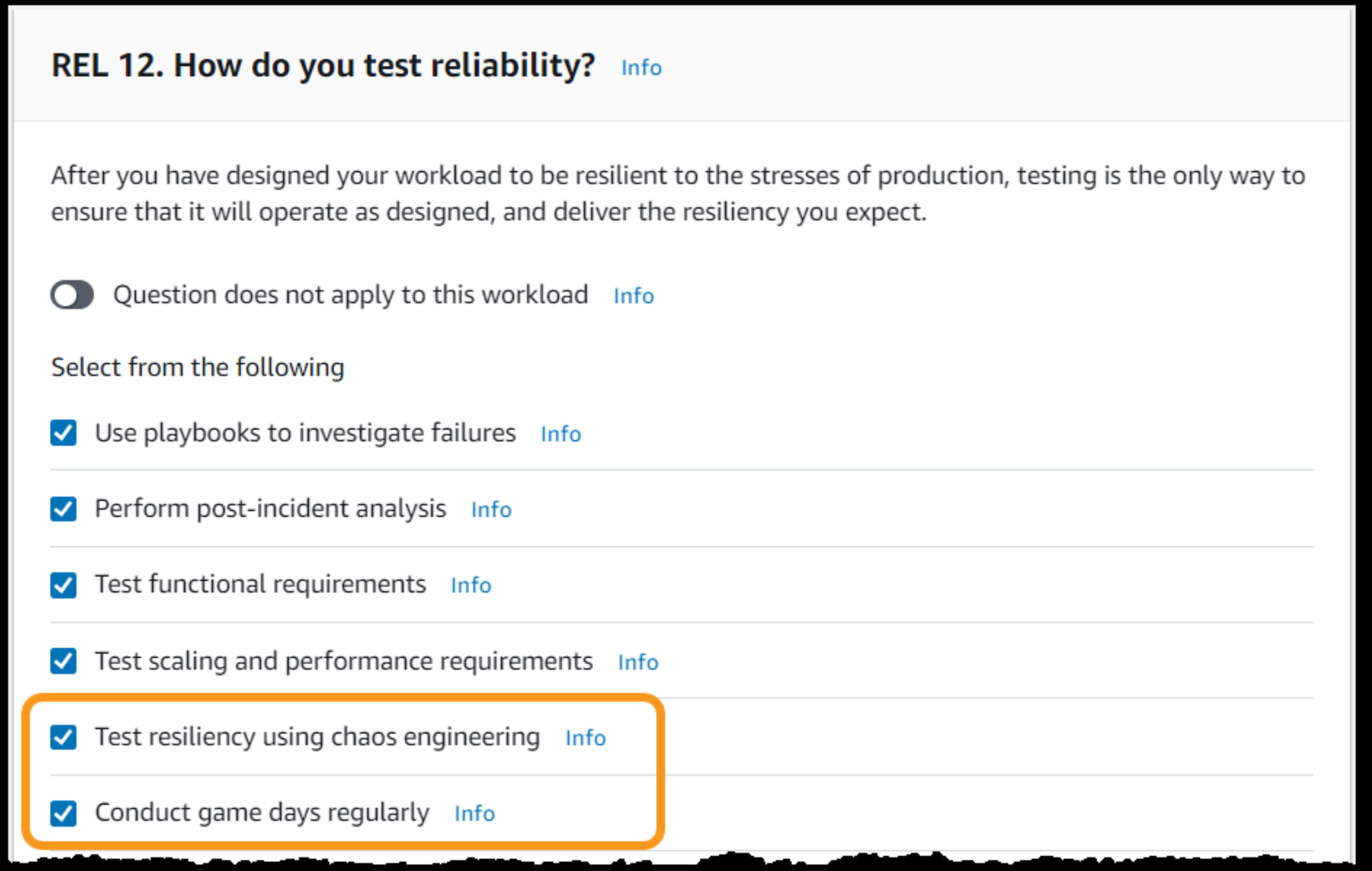


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next steps

You *can* do this

You *should* do this



Architecture resources

AWS Well-Architected Framework

The official best practices for Architecting in the AWS Cloud

aws.com/well-architected **(reliability pillar:** *bit.ly/reliability-pillar***)**

AWS Well-Architected Labs

Hands-on labs to help you learn, measure, and build using architectural best practices

wellarchitectedlabs.com **(reliability labs:** *wellarchitectedlabs.com/reliability***)**

AWS Architecture Center

Official AWS repository for all Architecture resources

aws.amazon.com/architecture

AWS Solutions Library

Vetted reference implementations and well-architected patterns

aws.amazon.com/solutions



AWS **Well-Architected**

Thank you!

Seth Eliot

Principal Reliability Solutions Architect, AWS Well-Architected
AWS
Twitter: @setheliot

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.